

Claims

- [c1] 1. A method for dynamically reducing latency over a communications network, the method comprising:
- a) determine completion of processing of a next data block;
 - b) determine a number of samples remaining in a queue of a consuming device of the next data block;
 - c) determine variability in the number of samples remaining in the queue of the consuming device; and
 - d) determine if the remaining number of samples in the queue of the consuming device can be reduced based upon the.
- [c2] 2. The method for dynamically reducing latency according to Claim 1, wherein the queue is reduced by increasing rate of consumption of the consuming device.
- [c3] 3. The method for dynamically reducing latency according to Claim 1, wherein the queue is reduced by resampling the samples to fewer samples.
- [c4] 4. The method for dynamically reducing latency according to Claim 1, wherein the queue is reduced by deleting samples in the queue.
- [c5] 5. The method for dynamically reducing latency according to Claim 1, wherein the queue is reduced by discarding samples of a data block.
- [c6] 6. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability.
- [c7] 7. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and an immediately preceding data block was added to the queue.
- [c8] 8. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in

the queue is greater than a sum of a number of samples in a data block and the variability and if at least one data block over a range of preceding data blocks was not added to the queue.

- [c9] 9. The method for dynamically reducing latency according to Claim 1, wherein samples in the queue are deleted if a total number of samples in the queue exceeds a predetermined threshold.
- [c10] 10. The method for dynamically reducing latency according to Claim 6, wherein the variability is weighted.
- [c11] 11. The method for dynamically reducing latency according to Claim 1, wherein the variability is weighted.
- [c12] 12. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block, the variability and a bias.
- [c13] 13. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and an immediately preceding data block was added to the queue.
- [c14] 14. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and if at least one data block over a range of preceding data blocks was not added to the queue.
- [c15] 15. The method for dynamically reducing latency according to Claim 13, wherein samples in the queue are deleted if a total number of samples in the queue exceeds a predetermined threshold.
- [c16] 16. The method for dynamically reducing latency according to Claim 12, wherein the variability is weighted.

- | | |
|-------|---|
| [c17] | 17. The method for dynamically reducing latency according to Claim 15, wherein the variability is weighted. |
| [c18] | 18. The method for dynamically reducing latency according to Claim 1, further including the step of repeating steps a) – d) for at least a portion of total data blocks. |
| [c19] | 19. The method for dynamically reducing latency according to Claim 18, wherein the queue is reduced by increasing rate of consumption of the consuming device. |
| [c20] | 20. The method for dynamically reducing latency according to Claim 18, wherein the queue is reduced by resampling the samples to fewer samples. |
| [c21] | 21. The method for dynamically reducing latency according to Claim 18, wherein the queue is reduced by deleting samples in the queue. |
| [c22] | 22. The method for dynamically reducing latency according to Claim 18, wherein the queue is reduced by discarding samples of a data block. |
| [c23] | 23. The method for dynamically reducing latency according to Claim 22, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability. |
| [c24] | 24. The method for dynamically reducing latency according to Claim 22, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and an immediately preceding data block was added to the queue. |
| [c25] | 25. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and if at least one data block over a range of preceding data blocks was not added to the queue. |

- [c26] 26. The method for dynamically reducing latency according to Claim 24, wherein samples in the queue are deleted if a total number of samples in the queue exceeds a predetermined threshold.
- [c27] 27. The method for dynamically reducing latency according to Claim 23, wherein the variability is weighted.
- [c28] 28. A method for dynamically reducing latency over a communications network, the method comprising:
- a) identifying when a data block is ready to be added to a queue of a consuming device;
 - b) polling the consuming device and calculating number of samples consumed by the consuming device since previous polling of the consuming device;
 - c) calculating variability in the number of samples consumed by the consuming device; and
 - d) determining if the remaining number of samples in the queue of the consuming device can be reduced based upon the variability.
- [c29] 29. A system for dynamically reducing latency over a communications network, the system comprising:
- a) means for determining completion of processing of a next data block;
 - b) means for determining number of samples remaining in a queue of a consuming device of the next data block;
 - c) means for determining variability in number of samples remaining in the queue of the consuming device; and
 - d) means for determining if the remaining number of samples in the queue of the consuming device can be reduced based upon the variability and if so reduce the queue.
- [c30] 30. The system for dynamically reducing latency according to Claim 29, wherein the queue is reduced by increasing rate of consumption of the consuming device.
- [c31] 31. The system for dynamically reducing latency according to Claim 29, wherein the queue is reduced by resampling the samples to fewer samples.

- [c32] 32. The system for dynamically reducing latency according to Claim 29, wherein the queue is reduced by deleting samples in the queue.
- [c33] 33. The system for dynamically reducing latency according to Claim 29, wherein the queue is reduced by discarding samples of a data block.
- [c34] 34. The system for dynamically reducing latency according to Claim 33, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability.
- [c35] 35. The system for dynamically reducing latency according to Claim 33, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and an immediately preceding data block was added to the queue.
- [c36] 36. The method for dynamically reducing latency according to Claim 5, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and if at least one data block over a range of preceding data blocks was not added to the queue.
- [c37] 37. The system for dynamically reducing latency according to Claim 31, wherein samples in the queue are deleted if a total number of samples in the queue exceeds a predetermined threshold.
- [c38] 38. The system for dynamically reducing latency according to Claim 34, wherein the variability is weighted.
- [c39] 39. The system for dynamically reducing latency according to Claim 33, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block, the variability and a bias.
- [c40] 40. The system for dynamically reducing latency according to Claim 29, further including the step of repeating steps a) – d) for a multiplicity of data blocks.

- [c41] 41. A software product for dynamically reducing latency over a communications network, the software product residing on a computer readable medium capable of instructing a processor to perform instructions to:
- a) determine completion of processing of a next data block;
 - b) determine number of samples remaining in a queue of a consuming device of the next data block;
 - c) determine variability in number of samples remaining in the queue of the consuming device; and
 - d) determine if the remaining number of samples in the queue of the consuming device can be reduced based upon the variability and if so reduce the queue.
- [c42] 42. The software product for dynamically reducing latency according to Claim 41, wherein the queue is reduced by increasing rate of consumption of the consuming device.
- [c43] 43. The software product for dynamically reducing latency according to Claim 41, wherein the queue is reduced by resampling the samples to fewer samples.
- [c44] 44. The software product for dynamically reducing latency according to Claim 41, wherein the queue is reduced by deleting samples in the queue.
- [c45] 45. The software product for dynamically reducing latency according to Claim 41, wherein the queue is reduced by discarding samples of a data block.
- [c46] 46. The software product for dynamically reducing latency according to Claim 45, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability.
- [c47] 47. The software product for dynamically reducing latency according to Claim 45, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and an immediately preceding data block was added to the queue.
- [c48] 48. The software product for dynamically reducing latency according to Claim

45, wherein samples of a data block are discarded if the number of samples remaining in the queue is greater than a sum of a number of samples in a data block and the variability and if at least one data block over a range of preceding data blocks was not added to the queue.

[c49] 49. The software product for dynamically reducing latency according to Claim 1, wherein samples in the queue are deleted if a total number of samples in the queue exceeds a predetermined threshold.

[c50] 50. A method for determining bias in an audio/video consuming device, the method comprising:
polling a consuming device for number of samples consumed between at time intervals; and
comparing number of samples consumed to a calculated number of samples that should have been consumed between the plurality of time intervals based on the consuming device set consumption rate.

[c51] 51. An audio and video consuming apparatus capable of dynamically reducing latency of data blocks received from a communications network, the apparatus comprising:
a processor, wherein the processor includes a memory;
peripherals in communication with and controlled by the processor, capable of at least one of sending, receiving and consuming data blocks; and
a software product, wherein the software product is capable of instructing the processor to execute instructions to:
determine completion of processing of a next data block;
determine a number of samples remaining in a queue of said consuming apparatus of the next data block;
determine variability in the number of samples remaining in the queue of said consuming apparatus; and
determine if the remaining number of samples in the queue of said consuming apparatus can be reduced based upon the variability.